Electronic Version
Stylesheet Version v1.1.1

# Description

# System and Methodology for Protecting New Computers by Applying a Preconfigured Security Update Policy

## Cross Reference to Related Applications

[0001] The present application is related to and claims the benefit of priority of the following commonly-owned, presently-pending provisional application(s): application serial no. 60/521,620 (Docket No. VIV/0018.00), filed June 7, 2004, entitled "System and Methodology for Protecting New Computers by Applying a Preconfigured Security Update Policy", of which the present application is a non-provisional application thereof. The present application is related to the following commonly-owned, presently-pending application(s): application serial no. 09/944,057 (Docket No. VIV/0003.01), filed August 30, 2001, entitled "System Providing Internet Access Management with Router-based Policy Enforcement"; application serial no. 10/159,820 (Docket No. VIV/0005.01), filed

May 31, 2002, entitled "System and Methodology for Security Policy Arbitration". The disclosures of each of the foregoing applications are hereby incorporated by reference in their entirety, including any appendices or attachments thereof, for all purposes.

## COPYRIGHT STATEMENT

[0002] A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

## BACKGROUND OF INVENTION

[0003] 1. Field of the Invention

[0004] The present invention relates generally to systems and methods for maintaining security of computer systems connected to one or more networks (Local Area Networks or Wide Area Networks) and, more particularly, to a system and methodology for securing newly acquired computers from security breaches by applying a preconfigured or preset security update policy.

[0005] 2. Description of the Background Art

[0006] The first computers were largely stand-alone units with no direct connection to other computers or computer networks. Data exchanges between computers were mainly accomplished by exchanging magnetic or optical media such as floppy disks. Over time, more and more computers were connected to each other using Local Area Networks or "LANs". In both cases, maintaining security and controlling what information a computer user could access was relatively simple because the overall computing environment was limited and clearly defined.

[0007] In traditional computing networks, a desktop computer largely remained in a fixed location and was physically connected to a single local network (e.g., via Ethernet). More recently, however, an increasingly large number of business and individual users are using portable computing devices, such as laptop computers, that are moved frequently and that connect into more than one network. For example, many users now have laptop computers that can be connected to networks at home, at work, and in numerous other locations. Many users also have home computers that are remotely connected to various organizations from time to time through the Internet. The num-

ber of computing devices, and the number of networks that these devices connect to, has increased dramatically in recent years.

[0008] In addition, various different types of connections may be utilized to connect to these different networks. A dial-up modem may be used for remote access to an office network. Various types of wireless connectivity, including IEEE (Institute of Electrical and Electronics Engineers) 802.11 and Bluetooth, are also increasingly popular. Wireless networks often have a large number of different users. Moreover, connection to these networks is often very easy, as connection does not require a physical link. Wireless and other types of networks are frequently provided in cafes, airports, convention centers, and other public locations to enable mobile computer users to connect to the Internet. Increasingly, users are also using the Internet to remotely connect to a number of different systems and networks. Thus, it is becoming more common for users to connect to a number of different networks from time to time through a number of different means.

[0009] One of the implications of this increasing number of devices occasionally connected to different networks is that traditional corporate firewall technologies are no longer

effective. Traditional firewall products guard a boundary (or gateway) between a local network, such as a corporate network, and a larger network, such as the Internet. These products primarily regulate traffic between physical networks by establishing and enforcing rules that regulate access based upon the protocol and type of access request, the source requesting access, the connection port to be accessed, and other factors. For example, a firewall may permit access to a particular computer using TCP/IP on TCP port 80, but deny remote access to other computers on the network. A firewall may also permit access from a specific IP address or range (or zone) of IP addresses, but deny access from other addresses. Different security rules may be defined for different zones of addresses. However, traditional firewall technology guarding a network boundary does not protect against traffic that does not traverse that boundary. It does not regulate traffic between two devices within the network or two devices outside the network. A corporate firewall provides some degree of protection when a device is connected to that particular corporate network, but it provides no protection when the device is connected to other networks.

[0010] One security measure that has been utilized by many

users is to install a personal firewall (or end point security) product on a computer system to control traffic into and out of the system. An end point security product can regulate all traffic into and out of a particular computing device. For example, an end point security product may expressly seek authorization from a user or administrator (or from a policy established by a user or administrator) for each network connection to or from a computing device, including connections initiated from the device and those initiated from external sources. This enables a user or administrator to monitor what applications on a device are accessing other machines or networks (e.g., the Internet). It also enforces security by obtaining authorization for each Internet or network connection opened to (or from) the device, including connections initiated both internally and externally. In the home environment, for instance, an end point security product enables a home user to monitor the applications he or she is using and enforces security by requiring his or her authorization for each connection. Typically, for connections initiated from the device, a user may configure application permission rules that permit certain applications to connect to one or more networks or devices, such as a local area network

(LAN) or a wide area network (WAN), such as the Internet. These application permission rules may, for instance, permit a particular application, such as a Web browser program, to open connections to the Internet. A rule may also be configured to permit an application to access another computer on the same LAN, but prohibit this application from opening an Internet connection.

[0011] Despite the increasing use of end point security and antivirus products, issues remain. Consumers currently face a particular problem when buying a new computer. Because of restrictions during the manufacturing process (e.g., due to cost/overhead issues, licensing restrictions, etc.), computers today tend to be outdated in terms of security by the time consumers actually have an opportunity to purchase those computers. For example, computers are frequently sold to consumers with an antivirus program already installed. However, the antivirus program and/or the virus definition files are typically out of date by the time the computer is actually received and placed into use by consumers. In order to update a computer for bringing it into compliance with current security updates, the user is required to connect the new computer to the Internet for accessing certain vendor sites, for example for obtain-

ing the latest antivirus definition file. Since a number of manufacturers update hard disk images for their computer lines only once or twice a year, a user may need to not only update data files (e.g., virus definition files) but also completely update the underlying security software itself, such as updating the underlying antivirus software (engine). Manufacturers' practice of annual or semi-annual updating is highly problematic. In terms of protection for a computer, that practice translates into a security system that may be up to 12 months out of date by the time the system actually gets into consumer hands.

[0012] Even if a consumer does everything exactly right with a new computer (e.g., updating antivirus software and data files, updating firewall software, updating operating system software, patching any applications with known vulnerabilities, etc.), he or she is required to spend a considerable amount of time online in order to get the "new" machine to a point where its security system is no longer out of date. For example, a new virus software update (e.g., from Symantec or McAfee) can easily run 15-20 MB to download. A new operating system service pack update (e.g., from Microsoft) may require a 100+ MB download. All told, the present day approach to delivering new com-

puters requires consumers to spend a considerable amount of time online with an outdated security system -- that is, a system which may have a long list of known vulnerabilities that hackers constantly scan for. As a concrete example from the inventor's own experience, a new notebook computer purchased while traveling was infected with the MS-Blast worm before even the brief task of downloading current firewall software (e.g., ZoneAlarm®, which is a fairly small download) could be completed.

[0013] To date, the only approach to addressing the foregoing is to preinstall antivirus and firewall/end point security software, as part of a computer's manufacturer-provided hard disk image. However as outlined above, with the current approach of manufacturing hard disk images, the preinstalled software is out of date by the time it actually reaches consumers. Accordingly, the foregoing problem of an initial infection has continued to plague consumers. Further compounding the problem, once a new machine has sustained an initial affection, the malicious software (e.g., virus, worm, etc.) can sabotage the machine, thus preventing the user from getting required downloads in order to bring the computer's security system up to date.

In other words, the initial infection prolongs the user's inability to get appropriate updates. Since malicious software often tends to be poorly written, infected machines tend to be prone to crashing. Although the failure comes from the infection, users may instead blame the computer manufacturer for a defective device: they bought a brand new machine and it failed, therefore it must be a defective machine. This leads to increased support/warranty costs and product returns for manufacturers, even though the failures are not necessarily a result of manufacturing defects.

[0014] What is needed is a solution for protecting newly purchased computers from viruses, worms, and other malicious software. The solution should protect the computer when it is initially received by the user and should facilitate the process of obtaining required updates in order to bring the computer's security system up to date. The present invention provides a solution for these and other needs.

## SUMMARY OF INVENTION

[0015] A system and methodology for protecting new computers by applying a preconfigured security update policy is described. In one embodiment, for example, a method of the

present invention is described for controlling connections to a computer upon its initial deployment, the method comprises steps of: upon initial deployment of the computer, applying a preconfigured security policy that establishes a restricted zone of preapproved hosts that the computer may connect to upon its initial deployment; receiving a request for a connection from the computer to a particular host; based on the preconfigured security policy, determining whether the particular host is within the restricted zone of preapproved hosts; and blocking the connection if the particular host is not within the restricted zone of preapproved hosts.

[0016] In another embodiment, for example, a computer system of the present invention that is preconfigured to control connections upon initial deployment is described that comprises: a computer having a preconfigured security policy that establishes a restricted zone of preapproved hosts that the computer may connect to upon initial deployment of the computer; a connectivity module for processing user requests for the computer to connect to a particular host; and a security module for determining whether the particular host is within the restricted zone of preapproved hosts based on the preconfigured security

policy, and for blocking any attempt to connect to a host that is not within the restricted zone of preapproved hosts.

[0017] In yet another embodiment, for example, a method of the present invention is described for enforcing pre-access connectivity restrictions on a new machine, the method comprises steps of: detecting attempts to connect the new machine to other devices; determining, based on an initial security policy that establishes a restricted zone of acceptable connections, which devices the new machine is permitted to connect to; and blocking any connection that attempts to connect the new machine to a device outside the restricted zone of acceptable connections.

## Brief Description of Drawings

[0018] Fig. 1 is a very general block diagram of a computer system (e.g., an IBM-compatible system) in which software-implemented processes of the present invention may be embodied.

[0019] Fig. 2 is a block diagram of a software system for controlling the operation of the computer system.

[0020] Fig. 3 is a block diagram of an environment in which the present invention is preferably embodied.

[0021] Fig. 4 is a flowchart illustrating the methodology of the

present invention for enforcing pre-access connectivity restrictions on a new machine.

[0022] Fig. 5 is a flowchart illustrating the operations of the system of the present invention in determining whether to permit access by an application during the restricted access stage.

## Detailed Description

### Glossary

[0023] The following definitions are offered for purposes of illustration, not limitation, in order to assist with understanding the discussion that follows.

[0024] End point security: End point security is a way of managing and enforcing security on each computer instead of relying upon a remote firewall or a remote gateway to provide security for the local machine or environment. End point security involves a security agent that resides locally on each machine. This agent monitors and controls the interaction of the local machine with other machines and devices that are connected on a LAN or a larger wide area network (WAN), such as the Internet, in order to provide security to the machine.

[0025] Firewall: A firewall is a set of related programs, typically

located at a network gateway server, that protects the resources of a private network from other networks by controlling access into and out of the private network. (The term also implies the security policy that is used with the programs.) A firewall, working closely with a router program, examines each network packet to determine whether to forward it toward its destination. A firewall may also include or work with a proxy server that makes network requests on behalf of users. A firewall is often installed in a specially designated computer separate from the rest of the network so that no incoming request directly accesses private network resources.

[0026] HTTP: HTTP is the acronym for HyperText Transfer Protocol, which is the underlying communication protocol used by the World Wide Web on the Internet. HTTP defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands. For example, when a user enters a URL in his or her browser, this actually sends an HTTP command to the Web server directing it to fetch and transmit the requested Web page. Further description of HTTP is available in "RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1", the disclosure of which is hereby incor-

porated by reference. RFC 2616 is available from the World Wide Web Consortium (W3C), and is available via the Internet (e.g., currently at www.w3.org/Protocols/). Additional description of HTTP is available in the technical and trade literature, see e.g., Stallings, W., "The Backbone of the Web", BYTE, October 1996, the disclosure of which is hereby incorporated by reference.

[0027] Network: A network is a group of two or more systems linked together. There are many types of computer networks, including local area networks (LANs), virtual private networks (VPNs), metropolitan area networks (MANs), campus area networks (CANs), and wide area networks (WANs) including the Internet. As used herein, the term "network" refers broadly to any group of two or more computer systems or devices that are linked together from time to time (or permanently).

[0028] RPC: RPC stands for remote procedure call, a type of protocol that allows a program on one computer to execute a program on another computer (e.g., a server computer). Using RPC, a system developer need not develop specific procedures for the server. The client program sends a message to the server with appropriate arguments and the server returns a message containing the results of the

program executed. For further description of RPC, see e.g., RFC 1831 titled "RPC: Remote Procedure Call Protocol Specification Version 2", available from the Internet Engineering Task Force (IETF), the disclosure of which is hereby incorporated by reference. A copy of RFC 1831 is available via the Internet (e.g., currently at www.ietf.org/rfc/rfc1831.txt).

[0029] Security policy: In general terms, a security policy is an organization's statement defining the rules and practices that regulate how it will provide security, handle intrusions, and recover from damage caused by security breaches. An explicit and well-defined security policy includes a set of rules that are used to determine whether a given subject will be permitted to gain access to a specific object. A security policy may be enforced by hardware and software systems that effectively implement access rules for access to systems and information. Further information on security policies is available in "RFC 2196: Site Security Handbook, (September 1997)", the disclosure of which is hereby incorporated by reference. A copy of RFC 2196 is available from the IETF via the Internet (e.g., currently at www.ietf.org/rfc/rfc2196.txt). For additional information, see also, e.g., "RFC 2704: The KeyNote Trust

Management System Version 2", the disclosure of which is hereby incorporated by reference. A copy of RFC 2704 is available from the IETF via the Internet (e.g., currently at www.ietf.org/rfc/rfc2704.txt). In this document, "security policy" or "policy" refers to a set of security policies and rules employed by an individual or by a corporation, government entity, or any other organization operating a network or other computing resources.

[0030] TCP: TCP stands for Transmission Control Protocol. TCP is one of the main protocols in TCP/IP networks. Whereas the IP protocol deals only with packets, TCP enables two hosts to establish a connection and exchange streams of data. TCP guarantees delivery of data and also guarantees that packets will be delivered in the same order in which they were sent. For an introduction to TCP, see e.g., "RFC 793: Transmission Control Program DARPA Internet Program Protocol Specification", the disclosure of which is hereby incorporated by reference. A copy of RFC 793 is available via the Internet (e.g., currently at www.ietf.org/rfc/rfc793.txt).

[0031] TCP/IP: TCP/IP stands for Transmission Control Protocol/Internet Protocol, the suite of communications protocols used to connect hosts on the Internet. TCP/IP uses several

protocols, the two main ones being TCP and IP. TCP/IP is built into the UNIX operating system and is used by the Internet, making it the de facto standard for transmitting data over networks. For an introduction to TCP/IP, see e.g., "RFC 1180: A TCP/IP Tutorial", the disclosure of which is hereby incorporated by reference. A copy of RFC 1180 is available via the Internet (e.g., currently at www.ietf.org/rfc/rfc1180.txt).

[0032] UDP: UDP stands for User Datagram Protocol, a connectionless protocol that, like TCP, runs on top of IP networks. Unlike TCP/IP, UDP/IP provides very few error recovery services, offering instead a direct way to send and receive datagrams over an IP network. UDP is used primarily for broadcasting messages over a network. For additional information on UDP, see RFC 768, "User Datagram Protocol", the disclosure of which is hereby incorporated by reference. A copy of RFC 768 is available via the Internet (e.g., currently at www.ietf.org/rfc/rfc768.txt).

[0033] URL: URL is an abbreviation of Uniform Resource Locator, the global address of documents and other resources on the World Wide Web. The first part of the address indicates what protocol to use, and the second part specifies the IP address or the domain name where the resource is

located.

[0034] Winsock: Windows Sockets 2 (Winsock) is a Microsoft-provided interface that enables programmers to create advanced Internet, intranet, and other network-capable applications to transmit application data across the wire, independent of the network protocol being used. With Winsock, programmers are provided access to advanced Microsoft Windows networking capabilities such as multicast and Quality of Service (QOS). Winsock follows the Windows Open System Architecture (WOSA) model; it defines a standard service provider interface (SPI) between the application programming interface (API), with its exported functions and the protocol stacks. It uses the sockets paradigm that was first popularized by Berkeley Software Distribution (BSD) UNIX. It was later adapted for Windows in Windows Sockets 1.1, with which Windows Sockets 2 applications are backward compatible. Winsock programming previously centered around TCP/IP. Some programming practices that worked with TCP/IP do not work with every protocol. As a result, the Windows Sockets 2 API adds functions where necessary to handle several protocols. For further information regarding Winsock, see e.g., "Winsock Reference", available from Microsoft

Corporation, the disclosure of which is hereby incorporated by reference. A copy of this documentation is available via the Internet (e.g., currently at msdn.microsoft.com/library/default.asp?url=/library/en-us/winsock/winsock/winsock_reference.asp).

## INTRODUCTION

[0035] Referring to the figures, exemplary embodiments of the invention will now be described. The following description will focus on the presently preferred embodiment of the present invention, which is implemented in desktop and/or server software (e.g., driver, application, or the like) operating in an Internet-connected environment running under an operating system, such as the Microsoft Windows operating system. The present invention, however, is not limited to any one particular application or any particular environment. Instead, those skilled in the art will find that the system and methods of the present invention may be advantageously embodied on a variety of different platforms, including Macintosh, Linux, Solaris, UNIX, FreeBSD, and the like. Therefore, the description of the exemplary embodiments that follows is for purposes of illustration and not limitation. The exemplary embodiments are primarily described with reference to block diagrams

or flowcharts. As to the flowcharts, each block within the flowcharts represents both a method step and an apparatus element for performing the method step. Depending upon the implementation, the corresponding apparatus element may be configured in hardware, software, firmware, or combinations thereof.

## COMPUTER-BASED IMPLEMENTATION

[0036] *Basic system hardware (e.g., for desktop and server computers)*

[0037] The present invention may be implemented on a conventional or general-purpose computer system, such as an IBM-compatible personal computer (PC) or server computer. Fig. 1 is a very general block diagram of a computer system (e.g., an IBM-compatible system) in which software-implemented processes of the present invention may be embodied. As shown, system 100 comprises a central processing unit(s) (CPU) or processor(s) 101 coupled to a random-access memory (RAM) 102, a read-only memory (ROM) 103, a keyboard 106, a printer 107, a pointing device 108, a display or video adapter 104 connected to a display device 105, a removable (mass) storage device 115 (e.g., floppy disk, CD-ROM, CD-R, CD-RW, DVD, or the like), a fixed (mass) storage device 116 (e.g.,

hard disk), a communication (COMM) port(s) or interface(s) 110, a modem 112, and a network interface card (NIC) or controller 111 (e.g., Ethernet). Although not shown separately, a real time system clock is included with the system 100, in a conventional manner.

[0038] CPU 101 comprises a processor of the Intel Pentium family of microprocessors. However, any other suitable processor may be utilized for implementing the present invention. The CPU 101 communicates with other components of the system via a bi-directional system bus (including any necessary input/output (I/O) controller circuitry and other "glue" logic). The bus, which includes address lines for addressing system memory, provides data transfer between and among the various components. Description of Pentium-class microprocessors and their instruction set, bus architecture, and control lines is available from Intel Corporation of Santa Clara, CA. Random-access memory 102 serves as the working memory for the CPU 101. In a typical configuration, RAM of sixty-four megabytes or more is employed. More or less memory may be used without departing from the scope of the present invention. The read-only memory (ROM) 103 contains the basic input/output system code (BIOS) -- a set of low-level rou-

tines in the ROM that application programs and the operating systems can use to interact with the hardware, including reading characters from the keyboard, outputting characters to printers, and so forth.

[0039] Mass storage devices 115, 116 provide persistent storage on fixed and removable media, such as magnetic, optical or magnetic-optical storage systems, flash memory, or any other available mass storage technology. The mass storage may be shared on a network, or it may be a dedicated mass storage. As shown in Fig. 1, fixed storage 116 stores a body of program and data for directing operation of the computer system, including an operating system, user application programs, driver and other support files, as well as other data files of all sorts. Typically, the fixed storage 116 serves as the main hard disk for the system.

[0040] In basic operation, program logic (including that which implements methodology of the present invention described below) is loaded from the removable storage 115 or fixed storage 116 into the main (RAM) memory 102, for execution by the CPU 101. During operation of the program logic, the system 100 accepts user input from a keyboard 106 and pointing device 108, as well as speech-based input from a voice recognition system (not shown).

The keyboard 106 permits selection of application programs, entry of keyboard-based input or data, and selection and manipulation of individual data objects displayed on the screen or display device 105. Likewise, the pointing device 108, such as a mouse, track ball, pen device, or the like, permits selection and manipulation of objects on the display device. In this manner, these input devices support manual user input for any process running on the system.

[0041] The computer system 100 displays text and/or graphic images and other data on the display device 105. The video adapter 104, which is interposed between the display 105 and the system's bus, drives the display device 105. The video adapter 104, which includes video memory accessible to the CPU 101, provides circuitry that converts pixel data stored in the video memory to a raster signal suitable for use by a cathode ray tube (CRT) raster or liquid crystal display (LCD) monitor. A hard copy of the displayed information, or other information within the system 100, may be obtained from the printer 107, or other output device. Printer 107 may include, for instance, an HP LaserJet printer (available from Hewlett Packard of Palo Alto, CA), for creating hard copy images of output of the

system.

[0042] The system itself communicates with other devices (e.g., other computers) via the network interface card (NIC) 111 connected to a network (e.g., Ethernet network, Bluetooth wireless network, or the like), and/or modem 112 (e.g., 56K baud, ISDN, DSL, or cable modem), examples of which are available from 3Com of Santa Clara, CA. The system 100 may also communicate with local occasionally-connected devices (e.g., serial cable-linked devices) via the communication (COMM) interface 110, which may include a RS-232 serial port, a Universal Serial Bus (USB) interface, or the like. Devices that will be commonly connected locally to the interface 110 include laptop computers, handheld organizers, digital cameras, and the like.

[0043] IBM-compatible personal computers and server computers are available from a variety of vendors. Representative vendors include Dell Computers of Round Rock, TX, Hewlett-Packard of Palo Alto, CA, and IBM of Armonk, NY. Other suitable computers include Apple-compatible computers (e.g., Macintosh), which are available from Apple Computer of Cupertino, CA, and Sun Solaris workstations, which are available from Sun Microsystems of Mountain View, CA.

[0044] *Basic system software*

[0045] Fig. 2 is a block diagram of a software system for controlling the operation of the computer system 100. As shown, a computer software system 200 is provided for directing the operation of the computer system 100. Software system 200, which is stored in system memory (RAM) 102 and on fixed storage (e.g., hard disk) 116, includes a kernel or operating system (OS) 210. The OS 210 manages low-level aspects of computer operation, including managing execution of processes, memory allocation, file input and output (I/O), and device I/O. One or more application programs, such as client application software or "programs" 201 (e.g., 201a, 201b, 201c, 201d) may be "loaded" (i.e., transferred from fixed storage 116 into memory 102) for execution by the system 100. The applications or other software intended for use on the computer system 100 may also be stored as a set of downloadable processor-executable instructions, for example, for downloading and installation from an Internet location (e.g., Web server).

[0046] System 200 includes a graphical user interface (GUI) 215, for receiving user commands and data in a graphical (e.g., "point-and-click") fashion. These inputs, in turn, may be

acted upon by the system 100 in accordance with instructions from operating system 210, and/or client application module(s) 201. The GUI 215 also serves to display the results of operation from the OS 210 and application(s) 201, whereupon the user may supply additional inputs or terminate the session. Typically, the OS 210 operates in conjunction with device drivers 220 (e.g., "Winsock" driver -- Windows' implementation of a TCP/IP stack) and the system BIOS microcode 230 (i.e., ROM-based microcode), particularly when interfacing with peripheral devices. OS 210 can be provided by a conventional operating system, such as Microsoft Windows 9x, Microsoft Windows NT, Microsoft Windows 2000, or Microsoft Windows XP, all available from Microsoft Corporation of Redmond, WA. Alternatively, OS 210 can also be an alternative operating system, such as the previously mentioned operating systems.

[0047] The above-described computer hardware and software are presented for purposes of illustrating the basic underlying desktop and server computer components that may be employed for implementing the present invention. For purposes of discussion, the following description will present examples in which it will be assumed that there exists connectivity of one device (e.g., desktop computer

or "client") to another (e.g., "server"). The present invention, however, is not limited to any particular environment or device configuration. In particular, a client/server distinction is not necessary to the invention, but is used to provide a framework for discussion. Instead, the present invention may be implemented in any type of system architecture or processing environment capable of supporting the methodologies of the present invention presented in detail below.

## PROTECTING NEWLY DEPLOYED COMPUTERS

[0048] *Introduction*

[0049] Classically, security has been divided in zones, such as a "trusted" zone (e.g., for one's own computer) and an "untrusted" zone (e.g., for the Internet). A "trusted zone" is a group of trusted computers defined by a user (or administrator) that is typically subject to less restrictive security rules than other computers and devices. For example, several computers in a home network may be included in the trusted zone of a user's security policy. Typically, all computers outside the trusted zone defined by the user are considered to be part of the "untrusted" zone. For example, remote machines accessible via the Internet are

generally "untrusted" and subject to more stringent security rules.

[0050] In accordance with the present invention, a new zone is introduced: a "restricted" zone (or "pre-access restricted zone") specifically for a new machine. Since the new machine operates in a restricted zone upon the initial deployment, the machine initially cannot be remotely accessed by another computer (e.g., a computer which is connected via a LAN or WAN). This restriction specifically addresses hacker probes, such as the MS-Blast worm. In the case of MS-Blast, a machine is not infected as a result of connecting to some malicious Web site or server. Instead, the MS-Blast worm infects machines by scanning open ports on machines and then delivering its malicious payload through a vulnerable port. In particular, the MS-Blast worm exploits a vulnerability of the RPC (Remote Procedure Call) service built into Microsoft Windows. The RPC service facilitates communication between applications and services over a network. The MS-Blast worm scans the local network for PCs that have port 135 open. If the worm finds such a target, it exploits the RPC vulnerability and infects the PC with a copy of itself. Once on a PC, the worm attempts to spread further and interfere

with normal OS operation. The worm also attempts to use infected computers in a distributed denial-of-service attack against Microsoft's Windows Update site.

[0051] *Overview to pre-access restricted zone*

[0052] In accordance with the present invention, when a manufacturer builds a hard disk image for a machine, the manufacturer places not only firewall and antivirus software on that image, but also sets up for the machine a set of pre-access firewall and access rules that limit the machine at the system level to only accessing specific sites (i.e., sites that the manufacture is aware of at the time that the image is built). In this manner, each machine receiving that image will be limited to only contacting a limited set of security-relevant sites (i.e., pre-access restricted zone). Importantly, all other attempted connections to the machine (i.e., from non-approved addresses) are refused during the pre- and peri-access stage. Only upon a given machine completing updating of security subsystems is the machine's security policy updated to allow other connections to occur. In particular, until the machine has updated relevant security components, the machine is not allowed to participate with general connectivity to the Internet, and the user is informed that is unsafe to do so

until the security-relevant updates have been completed. The user may be given the option to override this pre-access restriction, but in that case the user assumes responsibility for his or her actions. In such a case, for example, the system may display a disclaimer/warning dialog that records the user's acknowledgment to assume such responsibility. Using the approach of the present invention, when a new machine is first connected to a network with Internet connectivity, the machine does not participate in general connectivity but instead is only allowed to connect to relevant update sites, such as an antivirus update site, a firewall update site, an operating system update site, and other such sites for updating components that may require security-relevant updates/upgrades.

## SYSTEM COMPONENTS

[0053] Fig. 3 is a block diagram of an environment 300 in which the present invention is preferably embodied. As shown on Fig. 3, environment 300 includes a security setting (or zone configuration) user interface 310, a database 320, a security system 330, an operating system kernel 340 and a firewall 350. Security system 330 includes a zone configuration (security setting) data structure 331, a network

information data structure 332, an OS network information API 333, a TrueVector® engine 334, and a firewall API 335. Each of these components will now be described in more detail.

[0054] The zone configuration (security setting) user interface 310 is a configuration tool that enables a user or administrator to establish security settings and apply those settings to one or more machines or subnets. The zone configuration user interface 310 is connected to the security system 330. The zone configuration settings (or security settings) established for the current machine or network are stored in the zone configuration (security setting) data structure 331. The zone configuration settings, which comprise security settings or rules for particular networks or groups of machines, are also persistently stored in the database (or policy module) 320. In the currently preferred embodiment, the database 320 is a hierarchical object-oriented database. However, the database 320 could alternatively be a relational database, a file system, and/or any other form of persistent storage. The network information data structure 332 includes information about the network or networks to which a device is currently connected and also contains the profile of these networks. In-

formation regarding networks to which a device has been connected is persistently stored in the database 320.

[0055] The OS network information API 333 is an interface used to obtain network information from the operating system kernel 340. For example, the OS network information API 333 may be used to obtain an IP address of a particular adapter, or multiple IP addresses of devices on a particular subnet. The OS network information API 333 is also used to determine the MAC address of any router or other gateway device that is serving the local subnet. A MAC address is a unique identification number that is assigned by the manufacturer to a specific router or device. For example, when a router sends a packet to another router, the router transmitting the packet identifies itself by both an IP address and a MAC address. Each operating system provides some facility to discover network information, including IP and MAC addresses. The OS network information API 333 enables the security system 330 to utilize this underlying operating system facility to obtain network information that is required to detect and profile different networks. As described below, different operating systems provide different facilities for the provision of network information.

[0056] The TrueVector® engine 334 receives messages regarding events and uses event handlers to process and respond to these messages. The engine 334 also sends messages to other components, for example a message through the firewall API 335 to make a configuration change to the firewall 350. In the currently preferred embodiment, security and behavioral policy definition and enforcement (e.g., definition and enforcement of firewall, network access, and antivirus policies) are provided by the TrueVector® engine available from Zone Labs, Inc. and described in further detail in commonly-owned U.S. Patent No. 5,987,611, entitled "System and Methodology for Managing Internet access on a per Application basis for Client Computers Connected to the Internet", the disclosure of which is hereby incorporated by reference.

[0057] The TrueVector engine 334 acts as a supervisor module for enforcing a security policy (i.e., set of security rules). The TrueVector engine can be used to enforce a variety of different types of security policies or rules. These security policies may include application permission rules, such as a rule preventing access to particular resources by a particular application (e.g., a RealAudio player application "ra32.exe") or a rule permitting access to only administra-

tor or user-approved applications. Similarly, a policy or rule can be established requiring a particular application to have a verifiable digital signature. Apart from application-based rules, policies can be established on the basis of non-application activities or features. For example, rules can also be established on the basis of including and/or excluding access to particular Internet sites. These security policies can be customized by a user or administrator and a multitude of different types of policy rules can be established and enforced, as desired. Further information regarding the establishment and enforcement of security policies is provided in commonly-owned Application Serial No. 09/944,057 (Docket No. VIV/0003.01), filed August 30, 2001, entitled "System Providing Internet Access Management with Router-based Policy Enforcement", and in commonly-owned Application Serial No. 10/159,820 (Docket No. VIV/0005.01), filed May 31, 2002, entitled "System and Method for Security Policy Arbitration". The foregoing references are hereby incorporated by reference in their entirety, including any appendices or attachments thereof, for all purposes.

[0058] The firewall API 335 is used to enable dynamic configuration of the firewall 350. The firewall 350 is a firewall that

includes a means to configure IP address groups, which are used to specify trusted zones and other zones. For example, using the firewall API 335, a computer or device (or a group of computers and devices) can be added to a trusted zone maintained by the firewall 350 without having to change the security settings applicable to that trusted zone. Operation of the foregoing components for performing the methodology of the present invention is next described.

## DETAILED OPERATION

[0059] Fig. 4 is a flowchart 400 illustrating the methodology of the present invention for enforcing pre-access connectivity restrictions on a new machine. The following description presents method steps that may be implemented using processor-executable instructions, for directing operation of a device under processor control. The processor-executable instructions may be stored on a computer-readable medium, such as CD, DVD, flash memory, or the like. The processor-executable instructions may also be stored as a set of downloadable processor-executable instructions, for example, for downloading and installation from an Internet location (e.g., Web server).

[0060] The method of the present invention for enforcing pre-

access connectivity restrictions on a new machine configured with a disk image constructed in accordance with the present invention may be summarized as follows. At step 401, the machine performs its usual basic initialization for connectivity (e.g., TCP/IP, HTTP, and the like). At step 402, the machine may, as an optional step, download a "security update policy" to be applied during the restricted access stage; the download occurs using a preapproved (trusted) URL that has been preconfigured on the disk image. In that instance, the machine is preconfigured to only allow access to that specific site for downloading the security update policy. The advantage of this optional step is that it provides more flexibility during the restricted access stage. For example, over the span of a year (i.e., the lifespan of the disk image) the operating system vendor (e.g., Microsoft) may have changed the location of its update site; a downloadable security update policy may easily pick up such changes. In the event that optional step 402 is not employed, then the machine instead employs a predefined (imaged) policy that has been placed on the disk image.

[0061] At step 403, the security update policy (whether downloaded or imaged) is applied to the machine. In the cur-

rently preferred embodiment, the machine is restricted to only allow certain applications resident on the machine to connect to specific security-relevant sites that are specified in the security update policy (i.e., pre-access restricted zone). All other connections (e.g., from non-approved applications or processes, and/or to non-approved destinations) are denied. Optionally, the blocked connection may be redirected to a URL specified by the security update policy, for example to display a Web page to the user indicating what steps are required (namely, which updates are required) in order to fulfill the requirements of the security update policy. If desired, the redirection may execute a script that automatically guides the user through the required steps for updating the machine in accordance with the security update policy. Although the currently preferred embodiment focuses on security updates (thereby preventing infection/compromise of new machines), the security update policy may incorporate non-security tasks, such as updating a new machine to update for application bugs (whether or not they pose a security risk) that are now known at the time that the new machine is deployed. As indicated by step 404, once the machine has complied with the security update policy, the

"restricted zone" is lifted and the machine may participate in general Internet connectivity (typically, in accordance with default firewall access rules, such as provided by Zone Labs' ZoneAlarm® product).

[0062] In accordance with the present invention, any and all access is restricted until the machine is brought into compliance. One cannot predict the actual individual updates required to bring a new machine into compliance. What one can predict however are the site (in the case of a downloaded "updated" security update policy) or few sites (in the case of a predetermined security update policy) that one must visit in order to update the new machine in a manner that prevents infections/security breaches. By restricting new machines to only accessing and being accessible by trusted sites in the context of updating new machines, the approach of the present invention prevents infections which under prior art approaches are commonplace.

[0063] *Handling an attempt for access by an application*

[0064] Fig. 5 is a flowchart 500 illustrating the operations of the system of the present invention in determining whether to permit access by an application during the restricted access stage. As previously described, a security update

policy imaged on the hard disk during manufacturing or downloaded to the machine is applied to restrict access to the machine until security-related updates to the machine may be completed. A restricted zone is defined for the machine to only allow certain applications resident on the machine to connect to specific security-relevant sites.

[0065] The process for determining whether or not an application on the machine is to be permitted to access a particular site generally proceeds as described below. The following uses an example in a Windows environment; however a similar process is also applicable in other operating environments. The process commences at step 501 when a request for access to the Internet by a particular application is detected. In a Windows environment, for example, an application requesting Internet access typically connects via a Winsock (Windows socket) interface. At step 502, this request for access is intercepted and re-directed to the TrueVector engine. At step 503, when the request is received by the TrueVector engine, the engine determines that an application is requesting access to the Internet and attempts to identify the particular application making the request (e.g., from a unique fingerprint of the application).

[0066] After identifying the application, the TrueVector engine then determines whether or not to permit the access requested by the application at step 504. The TrueVector engine may consult a database or policy module to determine the policy applicable to this particular application. In the presently preferred embodiment, the policy database generally indicates if an application has been specifically approved for Internet access, has specifically been blocked (i.e., not approved for access), has not yet been evaluated, or is subject to a policy which provides for asking the user of the device whether or not to permit access. Currently, an application permission rule may be configured for one or more application(s) on a machine, that requires issuance of a notification (e.g., through issuance of an alert or prompt to the user interface) requesting a decision from the user as to whether or not to permit access by such application(s) to access a local (or trusted) zone and/or the Internet. In this case, however, the pre-defined security update policy specifies a restricted zone that provides that only certain specific applications are permitted to access a particular set of sites. Accordingly, the engine determines whether or not this particular application is one of the applications that is permitted to access the In-

ternet (e.g., for purposes of updating security on the machine). If the application is permitted to access the Internet, a check is made to determine if the site which the application is attempting to access is one that is allowed under the rules (security update policy). The IP address for the site which the application seeks to communicate with is compared against a list of allowed addresses. Certain sites can have multiple IP addresses. Accordingly, the system of the present invention currently stores the IP addresses with the respective Web sites, so that a particular site can be resolved at the level of its individual IP addresses. In this manner, the system of the present invention permits access based either on a Web site name (e.g., www.cnn.com) or based on a particular IP address.

[0067] If the particular application is approved for access and the requested site is on the list of allowed addresses, the application is permitted to access the site as provided at step 505. However, if the particular application is not one approved for access, or if the site is not on the approved list of addresses, then communication is blocked as provided at step 506. Generally, the application is blocked from connecting to the site until the security update process has been completed. Optionally, the blocked connec-

tion may be redirected to a URL specified in the security update policy for facilitating compliance with the security update policy as described above.

[0068] In an alternative embodiment, instead of blocking access which is prohibited by the security update policy, an event or alert may be sent to the user interface requesting a response from the user as to whether or not the requested access to the Internet should be permitted. The user may then be informed that is unsafe to permit access by the application until the security-relevant updates have been completed. However, the user may be given the option to override the restriction and allow access by the application.

[0069] While the invention is described in some detail with specific reference to a single-preferred embodiment and certain alternatives, there is no intent to limit the invention to that particular embodiment or those specific alternatives. For instance, those skilled in the art will appreciate that modifications may be made to the preferred embodiment without departing from the teachings of the present invention.